ストリーミング中の頻出アイテム発見アルゴリズム (計算機科学とアルゴリズムの数理的基礎とその応用)

AUTHOR(S):

緒方, 正虎; 来嶋, 秀治; 山下, 雅史

CITATION:

緒方, 正虎 ...[et al]. ストリーミング中の頻出アイテム発見アルゴリズム (計算機科学とアルゴリズムの数理的基礎とその応用). 数理解析研究所講究録 2011, 1744: 205-208

ISSUE DATE:

2011-06

URL:

http://hdl.handle.net/2433/170948

数理解析研究所講究録
第1744巻 2011年 205-208

2010年度冬のLAシンポジウム [S9]

ストリーミング中の頻出アイテム発見アルゴリズム

緒方正虎* 来嶋秀治* 山下雅史*

1 はじめに

与えられる集合より，ある閾値を越える要素集合を検知する問題がよく知られている．このような問題はiceberg問題と呼ばれ，我々が扱うべき情報量が増大するに伴い研究が進められてきた．[1][2][3]

iceberg問題は主にネットワーク監視やウェブクエリーの解析での応用が考えられている．代表的な例としてIPアドレスの解析によるDDos攻撃検知が挙げられる．ルーター等に流れる大量のIPアドレス中で頻出なIPアドレスは，DDos攻撃のターゲットとなっている可能性があり，これを検知する．

iceberg問題に対する基本的なアルゴリズムにKarpらの手法 [1] がある．この手法は単一ノード上に出現するアイテム集合から頻出アイテム候補の集合を出力する偽陽性アルゴリズムで，定数容量，定数時間で動作することが知られている．Karpらのアルゴリズムは記憶容量の点で非常に効率が良い．

一方，より実用に近いアルゴリズムも提案されている．Zahoらの手法 [2] ではサーバとノードの関係が1対多数のモデルを仮定している．この手法はKarpらの手法と異なり，各ノードの情報を集約し，確率的な手法を用いて頻出アイテムの数え上げを行う．

本研究では，Karpらのアルゴリズムに確率的手法を導入したアルゴリズムを設計する．いくつかのアルゴリズムを提案し，それぞれに対し確率的な下界を与える．

*九州大学 システム情報科学府

2 準備

2.1 頻出アイテムの定義

本研究，並びにKarpのアルゴリズムでは単一ストリームに要素集合から連続的にアイテムが入力されるモデルを考える．本研究，並びにKarpらのアルゴリズムで扱う頻出アイテムを以下の様に定義する．

定義 2.1 (頻出アイテム)．与えられる集合 S より，アイテム列 $x = (x_1, x_2, ..., x_N)$ が入力されたとする．S 中の任意のアイテム a の出現数を $f(a)$ とし，ユーザが設定する $\theta(0 < \theta < 1)$ とアイテム総数 N から閾値を $N\theta$ と定める．この時，頻出アイテムを以下の様に定義する．

$$\{a \in S \mid f(a) > N\theta\} \tag{1}$$

ここでパラメータ θ はアイテム総数 N に対する割合を意味する．

2.2 ナイーブな手法

Karpらは頻出アイテム発見問題のナイーブな手法の下界に関して触れている．[1] ここでのナイーブな手法とは，出現するアイテムの種類と数を全て保持し，その中から頻出アイテムを発見する手法を指す．定理を以下に示す．

定理 2.1 (頻出アイテム発見)．アルゴリズムに n 種類のアイテムが入力されたとする．頻出アイテム発見問題に対するどのようなオンラインアルゴリズムも最悪の場合 $\Omega(n \log(N/n))$ ビットの記憶容量を必要とする．

頻出アイテム発見問題では通常，アイテムの種類 n 並びにその総数 N は大きく，$n << N$ を想定するため，最悪の場合の記憶容量は大きくなる．

3 Karpらのアルゴリズム

Karpらのアルゴリズム [1] は単一ストリーム上の頻出アイテムを検知する偽陽性アルゴリズムで，与えられた集合から頻出アイテムの候補の集合を出力する．出力される集合は定義 2.1 を満たす頻出アイテムを必ず含むことが保証されている．アルゴリズムは一つのアイテムに対して線形時間，ユーザの定めた記憶容量で実行されることが知られている．以下に擬似コードを記す．

アルゴリズム 3.1 (Karpらのアルゴリズム).

```
x[1]...x[N] is the input sequence
K is a set of symbols initially empty
count is an array of integers indexed by K
  for i:= 1,...,N do
      {if x[i] is in K then
        count[x[i]] := count[x[i]] + 1

       else {insert x[i] in K,
             set count[x[i]] := 1

  if |K| > 1/theta  then
   for all a in K do
     { count[a] := count[a] - 1
     if count[a] = 0 then delete a from K}

      output K
```

到着したアイテムに対して K に含まれる要素の加算もしくは減算が必要であるが，これは $O(1)$ で計算が可能である．また集合 K は $\lfloor 1/\theta \rfloor$ の容量を必要とすることから，$O(1/\theta)$ である．このアルゴリズムは記憶容量，計算量の点で効率的である．一方，保持するアイテムを減算して頻出アイテムを絞り込むため，出現したアイテムの数え上げができない，文字列の順番により出力される K が変化する，という性質がある．

4 提案アルゴリズム

本研究ではKarpらのアルゴリズム [1] に確率的手法を導入する．入力要素を乱択することで，文字列の順序に依存せず K を出力する．アルゴリズムは入力文字列の要素をユーザが設定する確率 p で乱択し，集合 K に入れる．入力文字列の最後まで操作を繰り返した後，K に含まれる要素を頻出アイテムの候補として出力する．擬似コードを用いてアルゴリズムの詳細を述べる．

アルゴリズム 4.1.

```
x={x_1,x_2,...,x_N}
|K|=1/theta
    INPUT  x
    OUTPUT K
for(i<N and |K|<1/theta)do
    {x_i をKに入れる   with 確率 p
     x_i を捨てる      otherwise}

    OUTPUT K
```

上記のアルゴリズムは確率 p に従い要素を乱択する．文字列をすべて観測するか，もしくは $|K| \geq \frac{1}{\theta}$ となった時にアルゴリズムは停止する．

4.1 素朴な手法

まず要素を確率 $p = \frac{1}{N\theta}$ で乱択することを考える．$f(a) > N\theta$ を満たす頻出アイテムは期待値の上では K に含まれる．しかし，頻出アイテム以外のアイテムも選択される恐れがあるため，高い確率で $|K| \geq \frac{1}{\theta}$ となり，アルゴリズムが停止する．これを防ぐため

本手法ではKarpらの手法と比較して定数倍の記憶領域を確保する．アルゴリズムが保持するメモリを$\frac{c}{N\theta}$とした時，以下の定理が成り立つ．

定理 4.1. $x \in S$, $x = (x_1, x_2, ..., x_N)$とする．乱択する確率を$p = \frac{1}{N\theta}$，メモリ量をc/θとして，アルゴリズム *3.1* を用いる．この時$\{a \mid f(a) > N\theta\}$は以下の確率で頻出アイテム集合$K$に含まれる．

$$\Pr(a \in K) > 1 - e^{-1} - e^{-\frac{1}{3}\frac{(c-1)^2}{\theta}} \tag{2}$$

証明. 定理4.1を示すために以下の2つの補題を導く．

補題 4.1. $f(a) > N\theta$を満たす任意のアイテムaがKに含まれない確率は以下のように求められる．

$$\Pr(a \notin K||K| < \frac{1}{\theta}) < \frac{1}{e} \tag{3}$$

証明. $f(a) = N\theta$のアイテムaを考える．aが$N\theta$回アルゴリズムに入力され，一度も選ばれない確率は以下のように求められる．

$$\Pr(a \notin K||K| < \frac{1}{\theta}) < (1 - \frac{1}{N\theta})^{N\theta} \tag{4}$$

ここで右辺は$1/e$に近似が可能である．したがって補題4.1を得る． □

補題 4.2. 確率$p = \frac{1}{N\theta}$，メモリ量をc/θとした時，メモリKが$|K| > \frac{c}{\theta}$となる確率は以下のように求められる．

$$\Pr(|K| > \frac{c}{\theta}) < e^{-\frac{1}{3}\frac{(c-1)^2}{\theta}} \tag{5}$$

証明. 証明：総計N個の要素が入力されることを考える．それぞれが別種の要素だと仮定すれば，Kに$\frac{c}{\theta}$個以上入る確率の上界をchernoff上界から求めることができる．まず，以下のような確率変数X_iを導入する．

$$X_i = \begin{cases} 1 \text{ with } p = \frac{1}{N\theta} \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

確率変数の期待値を$E[X_i] = \frac{1}{N\theta}$とし，$K$に入る要素の期待値$X = \sum E[X_i]$を以下のchernoff不等式に適用する．

$$\Pr(X \geq (1+\delta)\mu) \leq e^{-\mu\delta^2/3} \tag{7}$$

補題4.2の式を得る． □

以上の二つの補題から定理4.1は証明される． □

4.2 パラメータの設定

定理4.1のパラメータでは頻出アイテムを見逃す確率が十分に小さいとは言えない．より高い確率で頻出アイテムを検知するには、確率pを適切に設定する必要がある．確率pを定数倍し，$p = \frac{t}{N\theta}$とする場合を考える．

定理 4.2. $x \in S$, $x = (x_1, x_2, ..., x_N)$とする．乱択する確率pを$p = \frac{t}{N\theta}$，メモリ量をc/θとし，アルゴリズム *4.1* を用いる．この時$c > t$とすれば$\{a \mid f(a) > N\theta\}$は以下の確率で頻出アイテム集合$K$に含まれる．

$$\Pr(a \in K) > 1 - e^{-t} - e^{-\frac{t}{3}\frac{(\frac{c}{t}-1)^2}{\theta}} \tag{8}$$

定理4.2を示すために以下の2つの補題を導く．

補題 4.3. $f(a) > N\theta$を満たす任意のアイテムaがKに含まれない確率は以下のように求められる．

$$\Pr(a \notin K||K| < \frac{1}{\theta}) < e^{-t} \tag{9}$$

証明. $f(a) = N\theta$のアイテムaが$N\theta$回アルゴリズムに入力され，一度も乱択されない確率は以下のように求められる．

$$\Pr(a \notin K||K| < \frac{c}{\theta}) < (1 - \frac{t}{N\theta})^{N\theta} \tag{10}$$

右辺を変形すれば，補題4.3を得る．

$$\begin{aligned} \Pr(a \notin K||K| < \frac{c}{\theta}) &< (1 - \frac{t}{N\theta})^{N\theta} && (11) \\ &< \left((\frac{\frac{N\theta}{t} - 1}{\frac{N\theta}{t}})^{\frac{N\theta}{t}} \right)^t && (12) \\ &< (e^{-1})^t && (13) \end{aligned}$$

□

補題 4.4. 確率$p = \frac{t}{N\theta}$，メモリ量をc/θとした時，メモリKが$|K| > \frac{c}{\theta}$となる確率は以下のように求められる．

$$\Pr\left(|K| > \frac{c}{\theta}\right) \leq e^{-\frac{t}{3}\frac{(\frac{c}{t}-1)^2}{\theta}} \tag{14}$$

証明. 定理 4.2 と同様の議論で証明を行う．$p = \frac{t}{N\theta}$ として確率変数 X_i を導入する，

$$X_i = \begin{cases} 1 \text{ with } p = \frac{t}{N\theta} \\ 0 \text{ otherwise} \end{cases} \tag{15}$$

総計 N 個の要素が入力されることを考える．それぞれが別種の要素だと仮定すれば，K に $\frac{c}{\theta}$ 個以上入る確率の上界を chernoff 上界から求めることができる．この時の期待値は $E[X_i] = \frac{t}{N\theta}$ であり，$X = \sum E[X_i]$ を以下の chernoff 不等式に適用すれば，補題 4.4 が求まる．

$$\Pr(X \geq (1+\delta)\mu) \leq e^{-\mu\delta^2/3} \tag{16}$$

□

以上の二つの補題から定理 4.2 が導かれる．

5 まとめ

本研究では Karp のアルゴリズムを基に，文字列の順序に対し独立な手法を提案した．しかし，Karp のアルゴリズムと比較して記憶容量が大きく改善が必要である．今後は確率的手法の特性を踏まえ，より効率的なアルゴリズム設計を目指す．

参考文献

[1] R. Karp, S. Shenker, and C. Papadimitriou, A simple algorithm for finding frequent elements in streams and bags, ACM Transactions on Database Systems, 28 (2003), 51–55.

[2] Q. Zhao, M. Ogihara, H. Wang, and J. Xu, Finding global icebergs over distributed data sets, in Proc. Symposium on Principles of Database Systems (PODS 2006), 298–307.

[3] 徳山豪，オンラインアルゴリズムとストリームアルゴリズム，共立出版, 2007.